

IN THE CLAIMS

Claims 10 through 21 were pending in this application. Claims 10,11 and 15 have been cancelled. Claims 12-14 and 16-21 have been amended. Claim 22 is new. All pending claims are reproduced below.

1 1-11. (Canceled).

1 12. (Currently Amended) The computer implemented method of claim ~~10~~ 22,
2 wherein ~~determining prior to run-time a resulting shape-tuple of the program expression~~
3 calculating, prior to run-time, a shape-tuple for the result of the program expression by creating a
4 shape-tuple expression comprising the shape-tuple for the operand and the shape-tuple operator
5 comprises the steps of:
6 determining a rank of the resulting shape-tuple; and,
7 promoting the ~~input~~ shape-tuple for ~~each~~ the operand to an appropriate rank.

1 13. (Currently Amended) The computer implemented method of claim 12, wherein
2 determining the rank of the resulting shape-tuple comprises the steps of:
3 ~~identifying~~ determining a rank of the ~~input~~ shape-tuple for ~~each~~ the operand;
4 identifying an operator corresponding to the operand ~~a built-in function in the program-~~
5 ~~expression; and~~
6 determining the rank of the ~~resulting~~ shape-tuple for the result of the program expression
7 according to the ~~built-in function~~ operator and the rank of the ~~input~~ shape-tuple for ~~each~~ the
8 operand.

1 14. (Currently Amended) The computer-implemented method of claim 12, wherein
2 promoting the ~~input~~ shape-tuple for ~~each~~ the operand to an appropriate rank comprises the steps
3 of:

4 comparing the rank of the ~~resulting~~ shape-tuple for the result of the program expression
5 to the rank of the ~~input~~ shape-tuple for ~~each~~ the operand;

6 responsive to the rank of the ~~resulting~~ shape-tuple for the result of the program
7 expression being greater than the rank of the ~~input~~ shape-tuple for ~~an~~ the operand, expanding the
8 ~~input~~ shape-tuple for the operand to correspond with the rank of the ~~resulting~~ shape-tuple for the
9 result of the program expression; and,

10 appending trailing extents of the expanded ~~input~~ shape-tuple for the operand with an
11 appropriate value.

1 15. (Canceled)

1 16. (Currently Amended) The computer-implemented method of claim ~~15~~ 22,
2 wherein ~~determining a shape-tuple operator for the built-in function~~ comprises the step of
3 mapping the program operator to an associated shape-tuple operator comprises:

4 looking up, in a table, a the shape-tuple operator corresponding to the ~~built-in function~~
5 program operator.

1 17. (Currently Amended) The computer-implemented method of claim ~~15~~ 22, further
2 comprising the step of calculating a shape predicate for the ~~resulting~~ shape-tuple for the result of
3 the program expression.

1 18. (Currently Amended) The computer-implemented method of claim ~~10~~ 22, further
2 comprising the steps of:

3 performing an array conformability check at run-time for a first ~~statement~~
4 program expression; and

5 applying a result of the conformability check to a second ~~statement~~ program
6 expression.

1 19. (Currently Amended) The computer-implemented method of claim 18, further
2 comprising the step of:

3 determining a relationship among the first ~~statement~~ program expression and the
4 second ~~statement~~ program expression.

1 20. (Currently Amended) The computer-implemented method of claim ~~10~~ 22, further
2 comprising the step of:

3 preallocating storage for each operand whose size is statically unknown, based
4 upon the ~~input~~ shape-tuple for each operand in a loop.

1 21. (Currently Amended) The computer-implemented method of claim 14, further
2 comprising:

3 responsive to the rank of the ~~resulting~~ shape-tuple for the result of the program
4 expression being less than the rank of the ~~input~~ shape-tuple for ~~an~~ the operand, truncating the
5 ~~input~~ shape-tuple for the operand corresponding with the rank of the ~~resulting~~ shape-tuple for the
6 result of the program expression.

1 22. (New) A computer-implemented method for inferring, prior to run-time, the array
2 shape of a result of a program expression of a high-level array-based language, the program
3 expression comprising an operand and a program operator, the method comprising:
4 arranging an extent for each dimension of the operand into a shape-tuple for the
5 operand;
6 identifying the program operator associated with the operand in the program
7 expression;
8 mapping the program operator to an associated shape-tuple operator, wherein the
9 shape-tuple operator is based upon the shape semantics of the program operator; and,
10 calculating, prior to run-time, a shape-tuple for the result of the program
11 expression by creating a shape-tuple expression comprising the shape-tuple for the
12 operand and the shape-tuple operator.